Санкт–Петербургский государственный университет

ДАВЫДОВ Дмитрий Владимирович

Выпускная квалификационная работа

Аспектно-ориентированный анализ тональности отзывов

Уровень образования: бакалавриат
Направление 02.03.02 «Фундаментальная информатика и информационные технологии»
Основная образовательная программа СВ.5003.2016 «Программирование и информационные технологии»
Профиль «Автоматизация научных исследований»

Научный руководитель:
доцент кафедры моделирования экономических систем,
кандидат физико-математических наук,
Ковшов Александр Михайлович

Санкт-Петербург
2020 г.

Содержание

Перечень сокращений и обозначений

Bi LSTM	Двунаправленная LSTM (Bidirectional LSTM)
CNN	Сверточная нейронная сеть
CRF	Условные случайные поля
DGL	Deep Graph Library
LSTM	Нейронная сеть с долгой краткосрочной памятью
RNN	Рекуррентная нейронная сеть
SVM	Метод опорных векторов
UD	Универсальные зависимости

Термины и определения

аспект или аспектная категория: Одна из сторон рассматриваемого объекта. Например, для ресторана аспектами могут быть качество еды, обслуживания, интерьер. Для автомобиля - комфорт, надежность, внешний вид и прочее.

аспектный термин: Последовательность слов, относящихся к заданному аспекту объекта. Аспектный термин извлекается из высказывания, выражающего некоторую точку зрения на одну из сторон объекта. Например, в предложении: *"Резюме следующее: место в целом отличное, цены не кусаются, но следует серьезно задуматься о замене персонала..."* в качестве аспектных терминов можно выделить: *"место"* (как относящееся к аспекту *ресторан-общее*) и *"персонала"* (как относящееся к аспекту *сервис-общее*).

грамматика: Раздел лингвистики, занимающийся изучением и описанием строения слов, словоизменения, видов словосочетаний и типов предложений. В нее входят морфология и синтаксис.

грамматические признаки: Самостоятельные части речи в русском языке имеют грамматические признаки. Например, существительные - род, склонение, число, падеж и др., глаголы - вид, наклонение, время и др.

контрольный алогоритм (baseline): Алгоритм, который используют для сравнения его результатов с результатами предложенных алгоритмов.

лексика: Словарный состав языка.

лексическая единица: Слово, устойчивое словосочетание или другая единица языка, способная обозначать предметы, явления, их признаки и т.п.

лемма: Нормальная форма слова. Например, единственное число, именительный падеж для существительных, неопределенная форма для глаголов.

мнение: Кортеж из трех элементов (аспектный термин, категория аспекта, тональность).

морфология: Часть грамматики, изучающая части речи, их категории и формы слов.

произведение Адамара: Поэлементное умножение.

самостоятельная часть речи: К самостоятельным частям речи относятся: существительное, прилагательное, числительное, местоимение, причастие, глагол, наречие, деепричастие.

сигмоида: Семейство гладких монотонно возрастающих функций. К этому семейству относятся логистическая функция $\sigma(x) = \frac{1}{1+e^{-x}}$ и гиперболический тангенс $tanh(x) = \frac{sh(x)}{ch(x)} = \frac{e^{2x}-1}{e^{2x}+1}$. Широко используются в нейронных сетях, так как производная легко выражается через саму функцию: $\frac{d\sigma}{dx} = (1-\sigma(x))\sigma(x)$, $\frac{dtanh}{dx} = \frac{ch^2(x)-sh^2(x)}{ch^2(x)} = 1 - tanh^2(x)$.

синтаксис: Раздел грамматики, описывающий правила, закономерности построения речи. В отличие от морфологии, рассматривающей отдельные слова и части слов, синтаксис рассматривает правила построения словосочетаний и предложений.

синтаксические связи: Синтаксические связи в русском языке делятся на сочинительные, подчинительные и координация.

токен: Значимая последовательность символов между разделителями. В качестве разделителей могут выступать пробельные символы, знаки пунктуации.

фонетика: Раздел лингвистики, изучающий звуковое строение языка.

Введение

В современной сети широко распространены различные платформы, где пользователи (как зарегистрированные так и анонимные) могут оставлять свои отзывы о товарах или услугах. Эта информация позволяет другим потенциальным покупателям при выборе между различными поставщиками товаров и услуг полагаться не только на их фактические характеристики, но и мнение других потребителей.

И для подавляющего большинства пользователей эта информация значима. Как показывают результаты опросов[1],[2] 9 из 10 покупателей признают, что пользовательский контент влияет на их решения о покупке. Такие результаты во многом связаны с тем, что техническая характеристика товара не дает представления об удобстве пользования им. В процессе экспуатации в нем могут быть выявлены значительные дефекты. Поэтому зачастую пользователь нацелено просматривает негативные отзывы (на большинстве платформ им соответствуют отзывы с низкими оценками), чтобы выявить недостатки товара и определить являются ли эти недостатки существенными для него.

Так как отзывы оказывает значительное влияние на решение о покупке потребителя, то они так же важны и для производителя. Отзывы служат механизмом обратной связи и показывают, какие особенности продукта нравятся пользователям, и потому их стоит оставить неизменными в дальнейшем развитии линейки товаров, а какие аспекты продукта, наоборот, необходимо дорабатывать.

Но многочисленность отзывов, отсутствие в них структуры не позволяет вручную извлекать из них интересующую информацию. Отсюда возникает необходимость в инструментах автоматического определения аспектных категорий объекта, выражений, характеризующих заданный аспект, и тональности обнаруженного выражения. Эти задачи являются подзадачами аспектно-ориентированного анализа тональности.

Эта задача была представлена в рамках международного семинара SemEval-2016 [**?**], посвященного семантическому анализу. Организаторами

соревнования был предоставлен набор отзывов[1] из определенных предметных областей (рестораны, автомобили, ноутбуки и т.д.). Для каждой предметной области был выбран набор аспектных категорий, которые представляют собой пару: сущность и атрибут (в таблице 1 строки представляют множество сущностей, а столбцы множество атрибутов. Если пара сущность-атрибут является аспектной категорией, то в соответствующей ячейке стоит галочка, иначе - крестик).

Таблица 1: Сущности и атрибуты аспектных категорий в отзывах о ресторанах.

Сущности \ Атрибуты	Общее	Цена	Качество	Оформление	Прочее
Обстановка	✓	✗	✗	✗	✗
Напитки	✗	✓	✓	✓	✗
Еда	✗	✓	✓	✓	✗
Расположение	✓	✗	✗	✗	✗
Ресторан	✓	✓	✗	✗	✓
Сервис	✓	✗	✗	✗	✗

В рамках семина задача рассматривалась в двух форматах: анализ на уровне предложений и всего отзыва. В данной работе рассматривается только первый формат.

Каждое упоминание в отзыве того или иного аспекта отмечалось экспертом в виде тройки: аспектный термин, категория аспекта, тональность. В дальнейшем будем называть такие тройки *мнениями*. Тональность могла быть представлена одним из четырех заначений: положительная, нейтральная, отрицательная и конфликт [2]. Например, в предложении: *"Резюме следующее: место в целом отличное, цены не кусаются, но следует*

[1]Данные предоставлены на различных языках, в рамках этой работы рассматриваются только отзывы на русском языке

[2]Последнее значение тональности выставлялось в том случае, если в предложении одна и таже аспектная категория указывалась как положительная, так и отрицательная. Например, с такой тональностью выражен аспект еда-качество в предложении: *"Шашлык из свинины был почти вкусным, если бы не попавшийся пересоленный кусочек (один из 5)."*

серьезно задуматься о замене персонала..." выделены следующие тройки: (место; ресторан-общее; положительный), (персонала; сервис-общее; отрицательный).

Задачу аспектно-ориентированного анализа тональности можно сформулировать как извлечение из предложения *мнений*. Тогда ее можно разделить на подзадачи:

1. Извлечение аспектных терминов - для полученного предложения выделить слова, посвященные какой-то из заданных аспектных категорий и определить какой[3].

2. Определение тональности - для заданного предложения, для которго указаны пары: категория аспекта и целевой объект. Требуется определить тональность каждой такой пары (положительная, нейтральная, негативная, конфликт).

[3]Данная подзадача в SemEval2016 называется Slot 12, так как объединяет в себе две другие Slot 1 и Slot 2.

Slot 1 - это задача определения какие из аспектных категорий, упомянуты в предложении.

Slot 2 - это задача извлечения аспектных терминов, но здесь не требуется указывать аспектную категорию.

Постановка задачи

Пусть T - алфавит языка. S - множество предложений. Имеется корпус документов разбитых на предложения: $\{s^{(1)}, \ldots, s^{(i)}, \ldots, s^{(n)}\}$. Предложение $s^{(i)}$, $i \in \{1, \ldots, n\}$ представляет собой последовательность $s^{(i)} = \{s_j^{(i)}\}_{j=1}^{k_i}$, каждый элемент которой является словом языка $s_j^{(i)} \in T$, $j = 1, \ldots, k$. Для последовательности $s^{(i)}$ будем обозначать множество ее подпоследовательностей как $\mathbf{s^{(i)}}$. $\bigcup\limits_i \mathbf{s^{(i)}} = \mathbf{S}$.

Предложения посвящены одной предметной области, для которой характерно множество аспектных категорий $A = \{a_1, \ldots, a_r, \ldots, a_m\}$, $A \subset T$. Множество тональностей $P = \{positive, neutral, negative\}$[4].

Как было сказано выше исходная задача по извлечению из предложения *мнений* т.е. $S \to (\mathbf{S},\ A,\ P)$ может быть разбита на подзадачи:

1. Извлечение аспектных терминов и определение их аспектной категории

Для неизвестной целевой зависимости f_1^*: $S \to (\mathbf{S}, A)$ известны значения на обучающей выборке $\{\{(s^{(i)}, \mathbf{s_j^{(i)}}, a_{r_j})\}_{j=1}^{i_m}\}_{i=1}^{n}$.

Необходимо построить алгоритм $f_1 = S \to (\mathbf{S}, A)$.

2. Определение тональности

Для неизвестной целевой зависимости f_2^*: $(S, \mathbf{S}, A) \to P$, известны значения на обучающей выборке $\{\{(s^{(i)}, \mathbf{s_j^{(i)}}, a_{r_j}, p_{q_j})\}_{j=1}^{i_m}\}_{i=1}^{n}$.

Необходимо построить алгоритм f_2: $(S, \mathbf{S}, A) \to P$.

[4]Как мог заметить читатель, в постановке задачи классификация происходит на три класса, тогда как ранее речь велась о четырех значениях тональности. Тональность *conflict* опущена, так как в обучающей выборке всего один аспектный термин с такой тональностью.

Глава 1. Обзор литературы

1.1 Извлечение аспектных терминов

Как предложено в работе [3] методы извлечения аспектных терминов можно классифицировать по подходам, на которых они основываются:

- Статистичекий
- Лингвистический
- Машинное обучение

1.1.1 Статистический

В работе статистических методов можно выделить несколько этапов:

1. Извлечение слов - кандидатов в аспектные термины.
2. Вычисление для них статистических характеристик.
3. Определение порога для статистичских характеристик, по которому будут отсеиваться неаспектные термины.

Одним из первых статистических методов был метод, основывающийся на выборе в качестве кандидатов наиболее часто использующихся существительных или фраз.

Однако такой подход упускает низкочастотные аспектые термины. Для того чтобы извлекать и низкочастотные слова предложена другая статистическая мера TF-IDF (1) (Term Frequency - частота слова, Inverse Document Frequency - обратная частота документа).

$$TF - IDF(t, d, D) = TF(t, d) \cdot IDF(t, D) \tag{1}$$

TF вычисляется как частота встречаемости термина в документе (2), где $count(t, d)$ - количество употреблений термина t в документе d, T_d -

множество терминов документа d.

$$TF(t,d) = \frac{count(t,d)}{\sum_{i \in T_d} count(i,d)} \tag{2}$$

Мера IDF вводится так, чтобы термины редко встречающиеся среди документов имели больший вес (3), где $|D|$ - число документов в корпусе, $|\{d \in D | t \in T_d\}|$ - число документов в корпусе, в которых встречается термин t.

$$IDF(t,D) = log \frac{|D|}{|\{d \in D | t \in T_d\}|} \tag{3}$$

1.1.2 Лингвистический

Лингвистический подход основывается на поиске аспектных терминов по заданным шаблонам. В качестве таких шаблонов могут выступать, например, последовательности существительных “голос солистки”, “подача блюд”, пары прилагательное-существительное “живая музыка”, “апельсиновый сок”, последовательности существительных и предлогов “блюда из мяса на мангале”и т.д.

Такие шаблоны могут быть сформированы заранее или построены автоматически. Например, в работе [4] определялись части речи слов и извлекались фразы, состоящие из двух слов, соответствующие шаблонам. Для автоматического конструирования шаблонов в тексте находятся термины близкие к аспектной категории, а затем рассматриваются их соседние слова и определяются их части речи, синтаксические связи. Конструкции, имеющие наибольшую частоту приниматся в качестве шаблонов.

Однако такой подход зачастую выделяет слова, не относящиеся к аспектным терминам. Для фильтрации кандидатов в аспектные термины можно использовать статистические характеристики.

Так же можно использовать набор правил для дополнения уже извлеченных терминов, например методами машинного обучения.

1.1.3 Машинное обучение

Задачу извлечения аспектных терминов можно рассматривать как задачу бинарной классификации слов на аспектные и неаспектные. В качестве признаков для классификаторов используют: часть речи, тип зависимости в синтаксическом дереве предложения (грамматика зависимостей рассматривается в 2.3.2 стр.21) и др.

В работе [5] из размеченной обучающей коллекции был сформирован список терминов для каждой аспектной категории. Затем вычислялась мера сходства слова с аспектной категорией (4) как сумма метрик сходства отдельных слов (5), где $\overrightarrow{w}$ - векторное представление слова w (векторные представления слов рассматриваются в 2.2 стр. 18), $\mathbf{a_i}$ - множество эталонных слов a_i аспекта.

$$simularity(w, a_i) = \sum_{s \in \mathbf{a_i}} simularity(w, s) \tag{4}$$

$$simularity(w, s) = \frac{\overrightarrow{w} \cdot \overrightarrow{s}}{||\overrightarrow{w}|| \cdot ||\overrightarrow{s}||} \tag{5}$$

Однако, таким образом будут извлечены однословные аспетные термины. Для того, чтобы извлекать аспектные термины, состоящие из нескольких слов в работе [5] использовался набор правил, которые объединяли однословные термины, стоявшие рядом, написанные через предлоги и др.

Задачу извлечения аспектов можно рассматривать как задачу разметки последовательности. Такой подход позволяет отличать одни аспектный термин, состоящий из нескольких слов, от нескольких подряд идущих. Существуют различные схемы кодирования последовательности.

- Например, участники SemEval-2015, использовавшие методы разметки последовательности, использовали BIO-кодирование [6]: B - начало аспектного термина, I - середина/конец аспектного термина, O - неаспектный термин.

- В работе [7] слова в аспекном термине подразделялись на главные и атрибуты. Для их кодирования использовались метки: FH - главное

слово, FA - предшествующий атрибут, FPA - последующий атрибут. Данный выбор обусловлен тем, что в проведенных авторами эксперементах точность извлечения аспектных терминов выше, если им присваиваются те же метки[5].

В качестве классификаторов в работах [7], [8] авторы использовали CRF [9]. В качестве контрольных алгоритмов часто используются SVM и наивный байесовский классификатор.

[5]В соответствии с предложенной схемой кодирования в выражениях *"компактная камера"* и *"камера компактная"* слово *камера* будет иметь метку FH. Тогда как в BIO-кодировании оно бы имело различные метки (I и B соответсвенно).

1.2 Определение тональности

Тональность по отношению к некоторому аспекту выражается посредством оценочных слов или целых фраз. Например, в качестве положительных оценочных слов могут выступать: *прекрасный, замечательный* и др., в качестве отрицательных: *ужасный, отвратительный* и др. Трудность определения оценочных слов состоит в том, что слова могут иметь различные значения. Для устранения этой многозначности необходимо знание контекста.

Например, некоторые слова оказываются оценочными только в некотором контексте. Например "*пресная* еда" (негативная тональность) и "*пресная* вода" (нейтральная тональность). Кроме того одно и тоже оценочное слово в одном контексте может быть положительным, а в другом иметь отрицательную окраску. Например, "*высокий* уровень обслуживания "и "*высокие* цены ". По этим причинам невозможно составить универсальный словарь, применимый для любой предметной области. Поэтому ряд работ предлагает методы автоматического формирования списков тональностей.

В работе [4] семантическая ориентация фразы вычисляется как (6) разность совместной встречаемости фразы со словами *"excellent"* (отличный) и *"poor"* (плохой) [6]. PMI вычисляется как (7), где $hits(query)$ - количество соответствий запросу $query$, $NEAR$ - оператор поисковой системы AltaVista, ограничивающий поиск 10 соседними словами.

$$SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor") \quad (6)$$

$$PMI(phrase_1, phrase_2) = log_2 \frac{hits(phrase_1\ NEAR\ phrase_2)}{hits(phrase_1)hits(prase_2)} \quad (7)$$

В работе [10] были составлены несколько корпусов, различающихся концентрацией оценочных слов и для наиболее высокочастотных слов вы-

[6] Были выбраны именно эти слова, потому что они обычно используются в рейтинговых системах: 5 баллов обозначаются как excellent, 1 балл как poor.

числены характеристики на этих корпусах: частотность, TF-IDF, отклонение от средней оценки и др. А затем построены бинарные классификаторы для автоматического разделения слов на оценочные и неоценочные.

В работе [5] были сформированы словари негативой и положительной тональностей. Затем для кандидатов в оценочные слова оценка тональности вычислялась двумя способыми:

1. Семантическая близость (4) к словарям.

2. Разность PMI между кандидатом и положительным и отрицательным классами. PMI, например для негативной тональности, вычислялась как (8), где $polarity(d)$ - тональность документа. В числителе дроби произведение количества документов в корпусе и количества сколько раз встречается терм t в документах негативной тональности. В знаменателе произведение сколько раз встречается терм в документах корпуса и количества терминов в отзывах негативной тональности.

$$PMI(t, negative) = log_2 \frac{\sum_{d \in \{d \in D | polarity(d) = negative\}} count(t, d)}{\sum_{d \in D} count(t, d)} \cdot \frac{|\bigcup_{\{T_d | d \in D\}}|}{|\bigcup_{\{T_d | d \in D \wedge polarity(d) = negative\}}|} \tag{8}$$

Затем для классификации использовался градиентный бустинг над решающими деревьями, а в качестве признаков - две полученные оценки тональности.

В работе [11] в качестве признаков использовались TF-IDF и часть речи и сравнивались результаты SVM и наивного баесовского классификатора.

Глава 2. Обработка данных

Текст является примером неструктурированных данных. Из него необходимо извлечь полезную информацию, сформировать признаки, которые затем будут использоваться классификаторами. В качестве признаков предложения в данной работе рассматриваются векторные представления его слов, сформированные на крупном корпусе документов и обладающие некоторыми полезными свойствами (подробнее в 2.2 стр. 18). Кроме того, для понимания семантики предложения определяется синтаксическая структура предложения.

Входные данные представляют собой набор размеченных отзывов в XML формате. Каждый отзыв это последовательность предложений, каждое из которых состоит из текста предложения и множества *мнений*. Для дальнейшей обработки предложения разбивались на токены с помощью регулярного выражения и хранились как список токенов, а аспектный термин в каждом *мнении*, как последовательность порядковых номеров токенов. Такой формат данных облегчал дальнейшую обработку данных, так как изменения отдельных токенов не требуют никаких изменений в *мнениях*. Модифицировать *мнения* требуется, только если в процессе обработки несколько токенов будут объединены в один, или наоборот один токен разделен на несколько. Такие действия производятся над токенами в процессе проверки орфографии.

2.1 Проверка орфографии

Отзывы в исходном наборе данных, как и многие другие отзывы в сети, содержат многочисленнные орфографические ошибки. Поэтому прежде чем приступить к выбору признаков необходимо провести предварительную обработку данных. Для проверки правописания использовался API Yandex.Speller[7]. Данный сервис исправляет различные орфографические ошибки в словах такие как[8]:

[7] https://yandex.ru/dev/speller/

[8] Далее примеры орфографических ошибок приводятся в следующем формате: *ошибочное написание* → *корректное написание.*

- неправильные буквы: *“рожденъе” → “рождение”*.
- пропущенные буквы: *“десер” → “десерт”*.
- лишние буквы: *“аттрибуты” → “атрибуты”*.
- неподходящее по контексту написание слов: *“прилично место” → “приличное место”*.
- и др.

Если спеллер нашел опечатку в тексте, то он указывает в какой последовательности символов она найдена и варианты исправлений. В качестве правильного написания использовалось первый вариант исправления. Исправления могут затрагивать как только один токен (*“месторасположения” → “месторасположения”*) так и несколько подряд идущих (*“не плохой” → “неплохой”*). Так же правильное написание может состоять из нескольких токенов (*“хорошос деланное”* и *“хорошо сделанное”*). В последних двух случаях изменяются не только токены, но и их индексы в *мнениях*.

2.2 Векторное представление слов

Под векторными представлениями понимается множество методов обучения представлениям, то есть автоматического формирования признаков для дальнейшего использования классификаторами. Как следует из названия они осуществляют отображение алфавита некоторого языка в векторное пространство $T \to R^m$. Векторные представления слов строятся таким образом, чтобы схожие по значению слова имели близкие представления (близость может вычисляться как она вычислялась в (5)). Достижение такого результата основывается на одноименной гипотезе дистрибутивной семантики. Она может формулироваться по разному: “слова похожие по смыслу встречаются в похожих контекстах”[14], “слова встречающиеся в схожих контекстах имеют схожий смысл”[15]. Так или иначе понятна идея дистрибутивной гипотезы: “существует зависимость между распределени-

ем слова в текстах и его значением, которая позволяет по первому сделать оценку второго"[16].

Под контекстом могут пониматься соседние слова в предложение, документе или целом корпусе документов. Для того чтобы отобразить контекст, в котором встречается слово, строится матрица совместной встречаемости слов. Для крупных корпусов документов размерность матрицы $X \in R^{n \times n}$ оказывается $n \approx 10^5$.

Затем к ней примененяются методы снижения размерности. Например, производится разложение на матрицы меньшей размерности как (9), где матрицы имеют размерности: $W \in R^{n \times m}$, $C \in R^{m \times n}$, m - размерность векторного пространства, в которое отображаются слова.

$$X = W \cdot C \tag{9}$$

Другой подход это применение нейросетей. В работе[17] предложено два метода полуечения векторного представления: Continuous Bag-of-Words и Continuous Skip-gram.

В модели Continuous Bag-of-Words производится проход по тексту скользящим окном фиксированного радиуса r и предсказание центрального слова по другим, попавшим в окно (10).

$$\sum_{w_i} P(w_i | w_{i-r}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+r}, W, C) \to max_{W,C} \tag{10}$$

В модели Continuous Skip-gram, наоборот, для каждого слова тренировочной выборки выбирается случайным образом радиус окна r и производится предсказание слов, попавших в окно, по центральному (11).

$$\sum_{w_i} P(w_{i-r}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+r} | w_i, W, C) \to max_{W,C} \tag{11}$$

В данной работе для векторного представления слов использовалась предобученная семантическая модель русского языка с сервиса RusVectōrēs.

Модель получена алгоритмом Continuous Skipgram, на корпусе "Тайга". Размер корпуса составляет почти 5 млрд слов, объем словаря 249 565 слов. Слова представляются в векторном пространстве размерностью 300. Слова хранятся в словаре в виде пары лемма слова и часть речи.

2.3 Синтаксическая структура предложения

Рассматривая естественный язык как сложный механизм, в нем традиционно выделяется по крайней мере три относительно независимых друг от друг компонента: фонетика, лексика и грамматика. Грамматика изучает закономерности построения осмысленной речи. По тому на каком уровне она исследуется, в грамматике выделяют две компоненты: морфологию, рассматривающую отдельные слова и части слов, и синтаксис, изучающий правила построения словосочетаний и предложений.

Перед синтаксисом стоят две основные задачи: теоритическая и описательная. Теоритическая - объяснение наблюдаемых в языке фактов. Описательная - формирование по этим фактам набора правил, определяющих грамматически правильные предложения. Правила выбираются таким образом, чтобы «добиться максимального совпадения множеств грамматически правильных и приемлимых (для носителей языка) предложений»[18].

Среди лингвистических теорий можно выделить два основных подхода к представлению синтаксической структуры предложения[19]: грамматика составляющих и грамматика зависимостей.

2.3.1 Грамматика составляющих

В грамматике составляющих всякая сложная единица текста целиком складывается из более простых не пересекающихся единиц — её непосредственно составляющих. Они в свою очередь так же делятся на более простые, вплоть до элементарных (неделимых)[9] [20]. Разделение проводится так, чтобы получившиеся части были максимально независимы друг от друга, т.е. имеют полноценное лексическое и грамматическое значение, самостоятельно употребляются вне данной конструкции.

[9]Обычно неделимыми состовляющими являются отдельные слова предложения.

Так как каждая составляющая (за исключением элементарных) представляет собой объединение соседних, в таком синтаксическом разборе предложения существенную роль играет порядок слов. Поэтому грамматика состовляющих используется для языков со строгим порядком слов, например английского.

Как указано в работе [21] выбор подходов отражает представления носителей языка. При отсутствии в языке развитого словоизменения важнейшим синтаксическим средством оказывается порядок слов. В то время как в языках, где слово представляется совокупностью форм, порядок слов оказывается менее значим. Соответственно для таких языков требуются другие подходы к синтаксическому разбору.

2.3.2 Грамматика зависимостей

Современная грамматика зависимостей в значительной степени основывается на идеях Л. Теньера[22]. На множестве слов предложения вводится бинарное антисимметричное отношение отношение зависимости. Синтаксическая структура предложения представляется в виде ориентированного дерева $G = (W, E)$ на множестве слов предложения W, ребрам которого E соответствуют отношения зависимости между словами. Для любого ребра дерева $(w_i, w_j) \in E$, $w_i, w_j \in W$ будем называть его начало w_i главным (head), а конец w_j зависимым (dependent).

Робинсон сформулировал аксиомы[23] дерева зависимостей:

1. $\exists! w_{root} \in W : \forall w_j \in W : \nexists (w_j, w_{root}) \in R$
   (Существует единственный независимый узел).

2. $\forall w_i \in (W \backslash w_{root}) : \exists! w_j : (w_j, w_i) \in R$
   (Для всех остальных узлов найдется единственный узел, для которого они зависимые).

3. $(w_{\mathbf{i}}, w_{\mathbf{j}}) \in R$, $i = min\{\mathbf{i}, \mathbf{j}\}$, $j = max\{\mathbf{i}, \mathbf{j}\}$ $=>$ $\forall k$: $i < k < j, \exists n : i \leq n \leq j, (w_n, w_k) \in R$
   (Если w_i зависит от w_j, то слово w_k, находящееся между ними в предложении, зависит от w_i, w_j или другого слова между ними.)

Последняя аксиома называется так же требованием проективности.

Хайс[24] и Гайфман[25] впервые рассмотрели математичексие свойства грамматики зависимостей. Однако в рассмотренной ими аксиоматизации деревья зависимостей были проективными.

Грамматика зависимостей, с возможностью строить непроективные деревья, позволяет строить деревья синтаксического разбора для языков с более свободным порядком слов чем в английском языке.[26]

2.4 Универсальные зависимости

В процессе анализа текста проводится грамматический разбор предложения. Для этого используются грамматические признаки и синтаксические связи, определяется роль лексической единицы в предложении. Но каждый язык имеет уникальный набор таких признаков, что затрудняет задачи межъязыкового анализа и разработки многоязыкового парсера.

Чтобы упростить эти задачи был создан открытый проект по формированию общей системы аннотаций зависимостей для различных языков. UD-проект объединил в один стандарт несколько уже существоваших схем[27]. Среди схем разметки зависимостей это стенфордская и схема зависимостей Google.

В таблице 2 изображена классификация универсальных зависимостей. Строки соответствуют категориям главных слов, а столбцы - зависимых.

Таблица 2: Классификация тегов универсальных зависимостей. Часть 1. (UD v2)

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp	-	-
Non-core dependents	obl vocative expl dislocated	advcl	advmod discourse	aux cop mark

Nominal dependents	nmod appos nummod	acl	amod	det clf case

В таблице 3 изображены универсальные зависимости, не являющиеся зависимостями в узком смысле.

Таблица 3: Классификация тегов универсальных зависимостей. Часть 2. (UD v2)

Coordination	Multiword expression	Loose	Special	Other
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

Участники проекта с использованием предложенного набора признаков разметили корпусы документов для различных языков. В последней, на данный момент времени, версии 2.5 (15 мая, 2019) представлено 157 размеченных корпусов для 90 языков[10]. Для русского языка размечено несколько корпусов документов. Один из них это первый аннотированный корпус текстов русского языка SynTagRus[11]. На нем обучена модель для русского языка в библиотеке StanfordNLP[28].

Данная библиотека содержит средства токенизации текста, определение частей речи, морфологичеких признаков, построение дерева зависимостей предложения. Именно для последней задачи используется библиотека. В процессе обработки текстов сохраняются часть речи и лемма слова, чтобы в дальнейшем получить по ним его векторное представление.

[10]https://universaldependencies.org/d

[11]https://universaldependencies.org/treebanks/ru_syntagrus/index.html

Глава 3. Рекуррентные нейронные сети

Как было указано в 1.2 для опредения тональности последовательности слов необходимо знание контекста, в котором они употребляются. Под контекстом можно понимать фиксированное количество соседних слов (например, рассматривать $n_{previous}$ предыдущих слов и $n_{subsequent}$ последующих слов). В такой интерпритации для классификации можно использовать CNN (сверточную нейронную сеть), на вход которой подавать векторные представления слов.

Такая трактовка контекста естественно является очень большим допущением. В некоторых текстах даже целого предложения недостаточно, чтобы определить тональность слова, необходимо знать так же и другие предложения текста[12]. Но, так как в данной работе рассматривается только анализ на уровне предложений, остановимся на допущении, что тональность термина можно определить по предложению, в котором оно употреблено.

Предложение представляет собой упорядоченный набор слов, и эта последовательность имеет произвольную длину. Для работы на таких данных широко применяются RNN (рекуррентные нейронные сети). RNN в отличие от сетей прямого распространения (многослойный перцептрон, CNN) имеют рекуррентные связи, т.е. обратные связи между нейронами одного или различных слоев. За счет них сети получают некоторое внутреннее состояние, в котором сохраняется и обновляется информация о контексте. Это внутреннее состояние отражает информацию о предыдущих элементах последовательности, причем количество этих элементов не фиксировано.

3.1 Простейшая RNN

Простейший пример рекуррентной нейронной сети это сеть, состоящая только из одного нейрона (рис. 1), выход которого (он называется

[12]Например, в предложении, взятом из обучающей выборки: *"Ну а о качестве и исполнении блюд даже нечего говорить!"* - тональность аспекта *качество еды* может интерпритироваться и как положительная и как отрицательная. Однако если знать, что предыдущим было предложение: *"Интерьер в легком французском стиле с приятным освещением на столиках, обслуживание без навязывания лишних блюд и напитков (официантка Анна), все происходило четко и быстро."* - то становится ясно, что аспектный термин имеет положительную тональность.

скрытым состоянием) поступает ему же на вход. Ее также можно представить развернутой по времени, где каждому моменту времени соответствует поступление очередного символа последовательности.

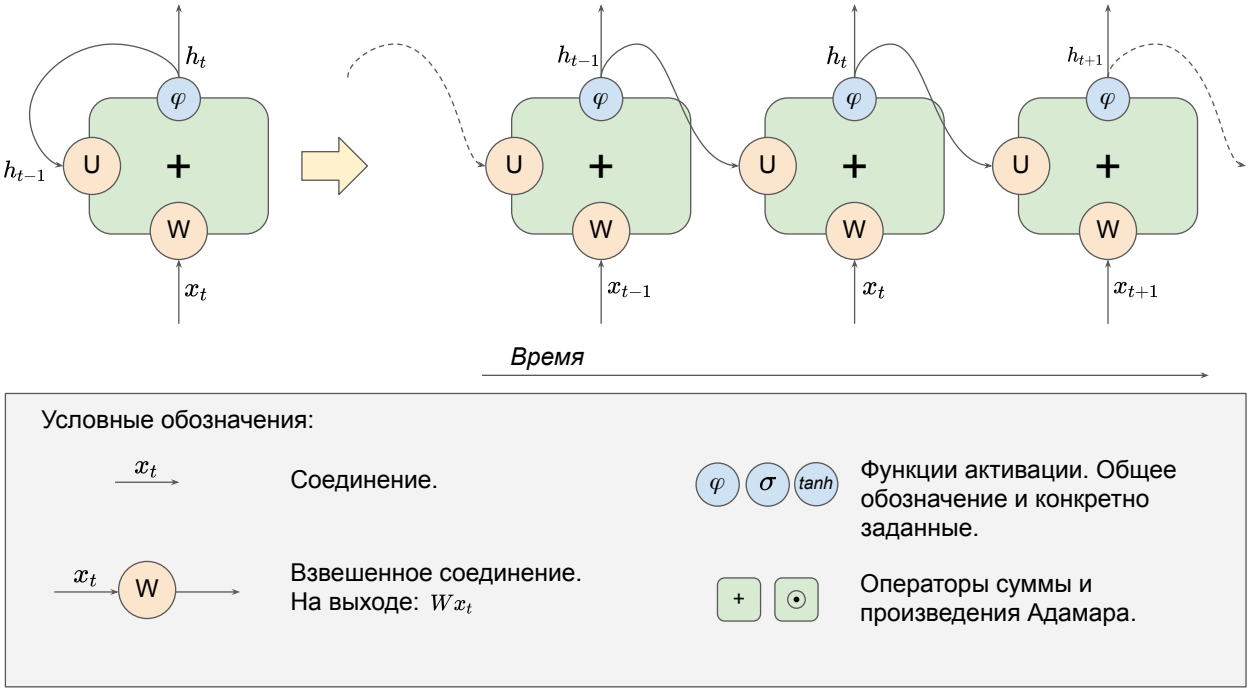

Рис. 1: Рекуррентная сеть из одного нейрона (слева). Эта же сеть развернутая по времени (справа).

Следующее скрытое состояние вычисляется по входному вектору и предыдущему скрытому состоянию как (12), где $x_t \in R^n$ входной вектор $t \in \{0, \ldots, T\}$ элемента, $h_{t-1} \in R^k$ - скрытое состояние $t-1$ элемента, $W \in R^{k \times n}$ - матрица весов для входного вектора, $U \in R^{k \times k}$ - матрица весов для скрытого состояния, $b \in R^k$ - вектор смещения, ϕ - функция активации. Заметим, что в каждый момент времени используются одни и те же веса.

$$h_t = \phi(Wx_t + Uh_{t-1} + b) \tag{12}$$

Пусть рассматривается задача классификации элементов последовательности, например определение тональности слов в предложении. То есть имеется последовательность $x_0, x_1, \ldots, x_T$ необходимо построить алгоритм, присваивающий ей последовательность меток $y_0, y_1, \ldots, y_T$. При-

менительно к рассмотренной выше простейшей рекуррентной нейронной сети, пусть имеется $f : R^k \to R^3$, необходимо найти оптимальные параметры рекуррентного слоя, т.е. минимизировать некоторую функцию потерь $L = \sum_{t=0}^{T} L_t(f(h_t), y_t)$.

Один из методов оптимизации параметров реккурентного слоя нейронной сети (рис. 1) это обратное распространение во времени. Частную производную функции потерь по параметру U можно записать как (13). Так как h_t явно зависит от параметра U, а так же посрадством параметра h_{t-1}, то частная производная по параметру U равна (14), где $\frac{\partial^+ h_i}{\partial U}$ обозначает частную производную h_i по U, а h_{i-1} принимается за константу. Аналогично расписывается производная по W.

$$\frac{\partial L}{\partial U} = \sum_{t=0}^{T} \frac{\partial L_t}{\partial U} \tag{13}$$

$$\begin{aligned}
\frac{\partial L_t}{\partial U} =& \frac{\partial L_t}{\partial h_t}\frac{\partial^+ h_t}{\partial U} + \\
& \frac{\partial L_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial^+ h_{t-1}}{\partial U} + \\
& \dots \\
& \frac{\partial L_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial h_{t-2}} \cdots \frac{\partial h_1}{\partial h_0}\frac{\partial^+ h_0}{\partial U} = \\
& \sum_{i=0}^{t} \frac{\partial L_t}{\partial h_t} \prod_{j=i+1}^{t} \left(\frac{\partial h_j}{\partial h_{j-1}}\right) \frac{\partial^+ h_i}{\partial U} = \\
& \sum_{i=0}^{t} \frac{\partial L_t}{\partial h_t}\frac{\partial h_t}{\partial h_i}\frac{\partial^+ h_i}{\partial U}
\end{aligned} \tag{14}$$

Теперь, вычислив градиенты, можно обновить переметры модели (15), где α - коэффициент скорости обучения.

$$U := U - \alpha \frac{\partial L}{\partial U} \tag{15}$$

Для RNN на последовательностях большой длины происходит взрыв или затухание градиента. Для борьбы с этой проблемой была предложена LSTM.

3.2 LSTM

LSTM (сеть с долговременной и кратковременной памятью) была предложена в работе [30]. В ней для учета значений как на коротких так и на больших промежутках времени используется ячейка памяти. Обновление ее состояния происходит посредством различных фильтров: входного, выходного, а так же фильтра забывания (последнего не было в работе [30]).

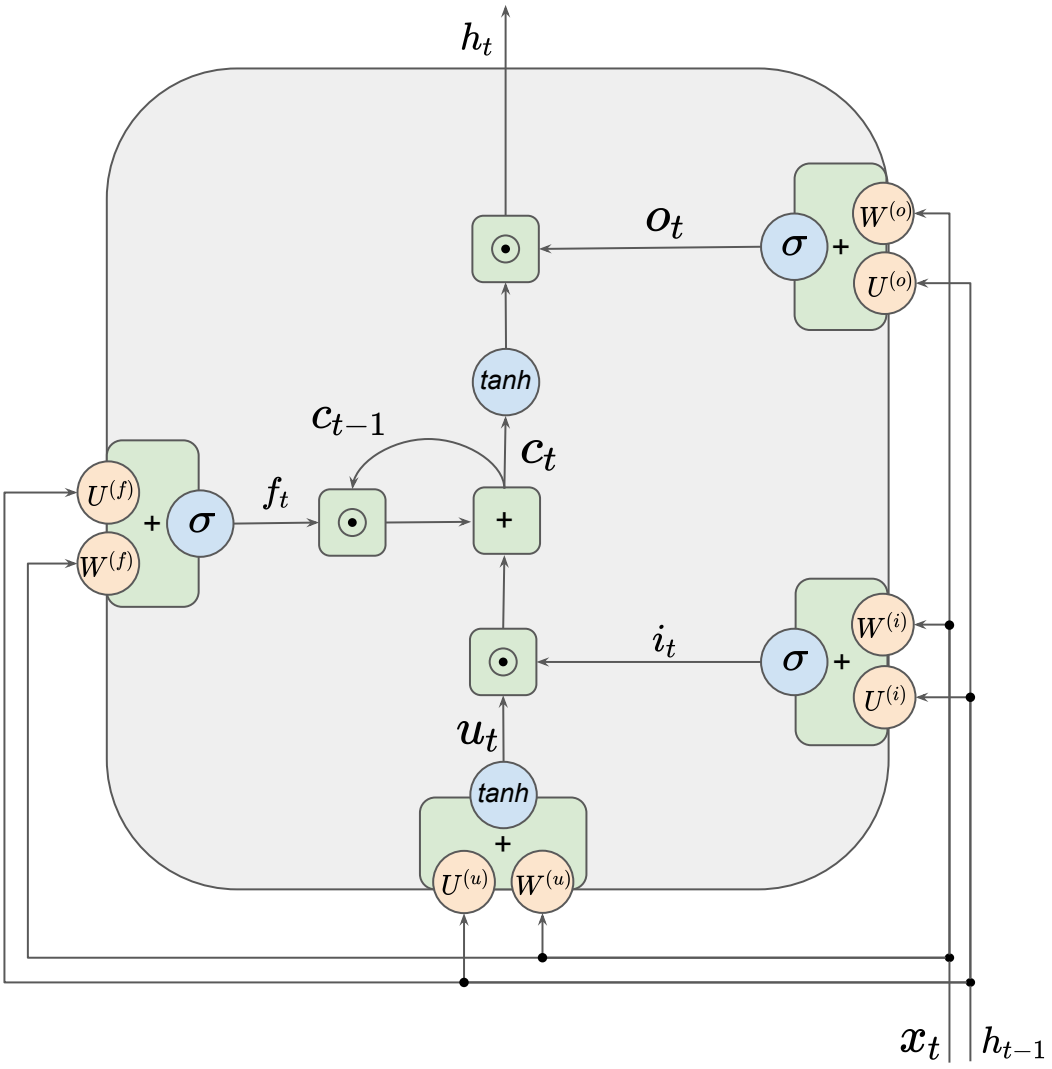

Рис. 2: Структура LSTM слоя.

Вектор новых значений ячейки памяти вычисляется как (16). Входной фильтр (17) возвращает числа в интервале $[0; 1]$ и определяет какую долю новых значений ячейки памяти пропустить дальше.

$$u_t = tanh(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}) \tag{16}$$

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \tag{17}$$

Аналогичную роль выполняет фильтр забывания, он определяет какие значения ячейки памяти сохранятся (18).

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}) \tag{18}$$

Таким образом, значение ячейки памяти вычисляются как (19).

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \tag{19}$$

К полученным значениям ячейки памяти применяется функция гиперболического тангенса. Наконец, что определить какие значения будут поданы на выход используесят выходной фильтр (20). Таким образом, получаем следущее скрытое состояние (21).

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}) \tag{20}$$

$$h_t = o_t \odot tanh(c_t) \tag{21}$$

В рассмотренных выше формулах $W^{(i)}$, $W^{(u)}$, $W^{(f)}$, $W^{(o)} \in R^{k \times n}$ - матрицы весов для входного вектора, $U^{(i)}$, $U^{(u)}$, $U^{(f)}$, $U^{(o)} \in R^{k \times k}$ - матрицы весов для скрытого состояния; $b^{(i)}$, $b^{(u)}$, $b^{(f)}$, $b^{(o)} \in R^k$ - вектора смещений; $\odot$ - произведение Адамара, σ - логистическая функция.

3.2.1 Bi LSTM

Однако, например, для задачи классификации слов предложения нужна информация не только о предыдущих, но и последующих словах. Поэтому широкое распространение получила Bi LSTM.

Эта нейронная сеть состоит из двух слоев LSTM сетей, одна из которых проходит по последовательности в прямом порядке, а другая в обрат-

ном (3). За счет этого Bi LSTM учитывает прошлый и будущий контекст слова.

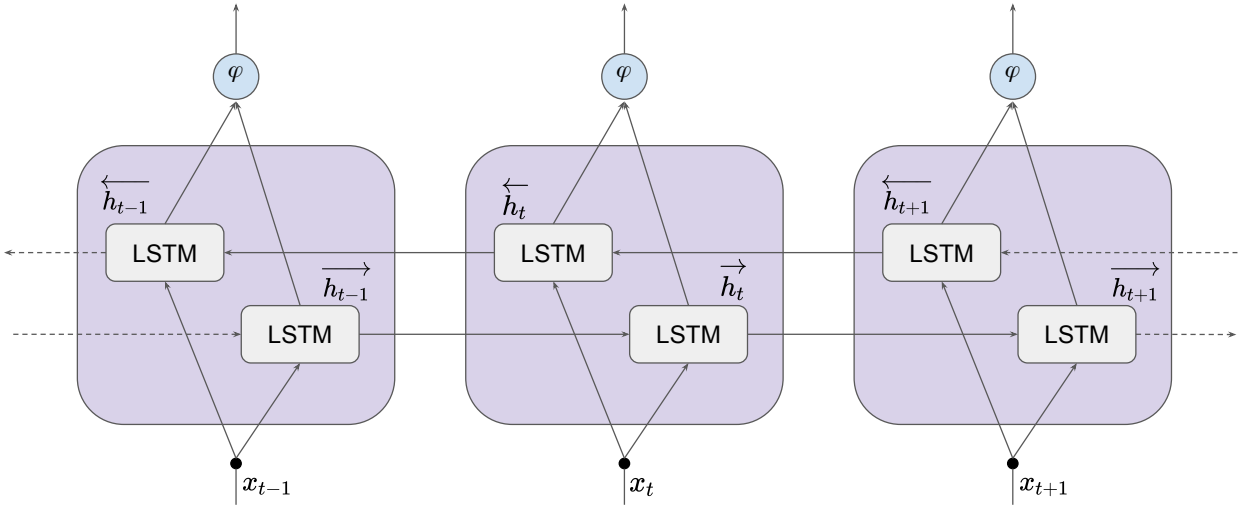

Рис. 3: Схема работы Bi LSTM.

3.2.2 Tree LSTM

В отличие от LSTM, которая на каждом шаге использует скрытое состояние предыдущего шага, Tree LSTM использует скрытые состояния всех дочерних узлов. Схематично ее работу можно отобразить (4).

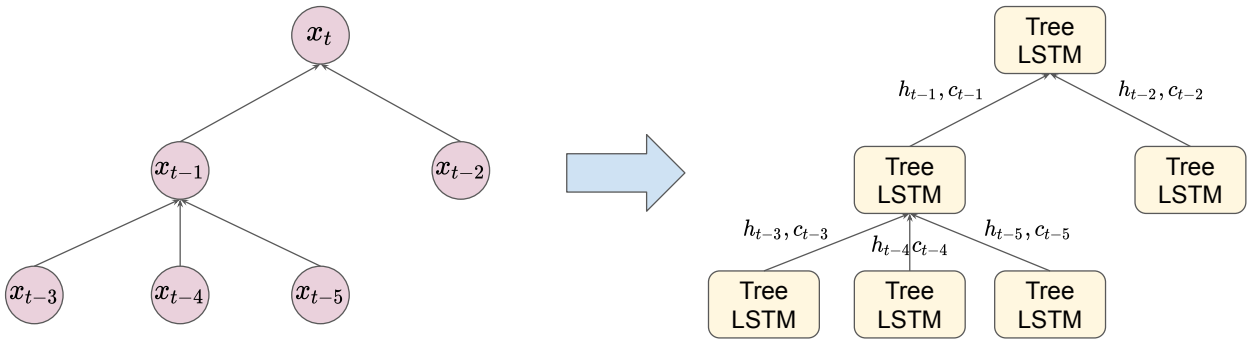

Рис. 4: Схема работы Tree LSTM.

Существуют различные вариации Tree LSTM, в данной работе использовалась Child-Sum Tree LSTM, т.е. Tree LSTM с суммированием скрытых состояний дочерних узлов (22), где $Q(t)$ - множество дечерних узлов узла t.

$$\widetilde{h}_t = \sum_{j \in Q(t)} h_j \tag{22}$$

Значения фильтров входа, выхода и нового состояния памяти вычисляются как и для LSTM, только вместо скрытого состояния предыдущего узла выступает сумма дочерних (23), (24), (25)

$$i_t = \sigma(W^{(i)} x_t + U^{(i)} \widetilde{h}_{t-1} + b^{(i)}) \tag{23}$$

$$u_t = tanh(W^{(u)} x_t + U^{(u)} \widetilde{h}_{t-1} + b^{(u)}) \tag{24}$$

$$o_t = \sigma(W^{(o)} x_t + U^{(o)} \widetilde{h}_{t-1} + b^{(o)}) \tag{25}$$

Значения фильтра забывания вычисляются для каждого дочернего узла (26).

$$f_{tq} = \sigma(U^{(f)} h_q + b^{(f)}) \tag{26}$$

Новое состояние ячейки памяти вычисляется как (27).

$$c_t = i_t \odot u_t + \sum_{q \in Q(t)} f_{tq} \odot c_q \tag{27}$$

Таким образом, получаем скрытое состояние узла t (28).

$$h_t = o_t \odot tanh(c_t) \tag{28}$$

Глава 4. Извлечение аспектных терминов

Отношение к некоторому аспекту объекта может выражаться без явного упоминания относящихся к нему слов. В таких случаях в обучающей выборке вместо слов аспектного термина указано *NULL*. Для того, чтобы извлекать как явные, так и неявные аспектные термины используется два классификатора на основе Bi LSTM.

4.1 Аспектные категории предложения

Первый классификатор определяет какие аспектные категории упоминаются в предложении. Так как в предложении могут быть выражены несколько аспектных категорий или нет ни одной, это задача классификации на пересекающиеся классы. Для ее решения использовалась нейронная сеть, реализованная с помощью библиотеки PyTorch.

Классификатор принимает на вход последовательность предложений и разбивает ее на батчи. По полученному батчу строится тензор из векторных представлений слов. Он проходит через слой дропаута и подается на вход Bi LSTM. Полученные на выходе из нее скрытые состояния, соответствующие последним словам предложений батча, подаются в полносвязный слой, осуществляющий линейное отображение (29), $A \in R^{[n \times 2m]}$, n - количество аспектных категорий предметной области, m - размерность скрытого состояния Bi LSTM.

$$y = Ah + b \tag{29}$$

К полученному таким образом вектору y применяется логистическая функция и по нему определяются аспектные категории предложения. Если значение вектора y_i больше порога для i аспектной категории, то считается, что она присутствует в предложении. Оптимальный порог выбирается методом Нелдера — Мида.

Для обучения нейронной сети в качестве функции потерь используется среднее бинарной перекрестной энтропии (30), где $y_i^* \in \{0, 1\}$ принимает значение 1, если i-ый аспект присутствует, иначе 0. $y_i \in [0, 1]$ - предсказан-

ное значение.

$$loss(y, y^*) = \sum_{i=1}^{n} \frac{y_i^* log(1 - y_i) + (1 - y_i^*) y_i}{n} \tag{30}$$

В качестве метрики качества модели использовалась f1 мера. Она вычислялась как (31), где *precision* - это отношение количества верно распознанных аспектов к количеству предсказаний классификатора, а *recall* - это отношение количества верно распознанных аспектов к количеству аспектов на самом деле.

$$f1 = 2\frac{precision * recall}{precision + recall} \tag{31}$$

Для выбора оптимальных параметров модели (количество эпох, размерность вектора скрытых состояний) здесь и в описанных далее классификаторах использовалась перекрестная проверка с разделением на 5 частей. Исходное множество предожений 5 раз разделялось на обучающую и валидационную выборки в соотоношении 4:1, в результате чего все 5 частей исходного множества используются для оценки качества модели (5). Полученные метрики качества усредняются.

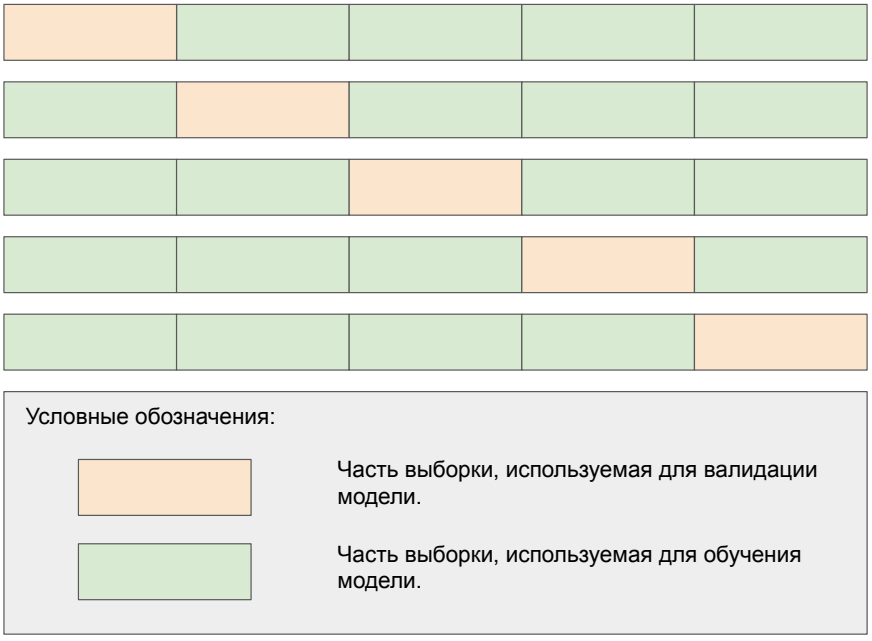

Рис. 5: Схема перекрестной проверки на 5 частей.

На рис. 6 изображена зависимость f1-меры к эпохе обучения классификатора. В качестве оптимального количества эпох выбрано 40.

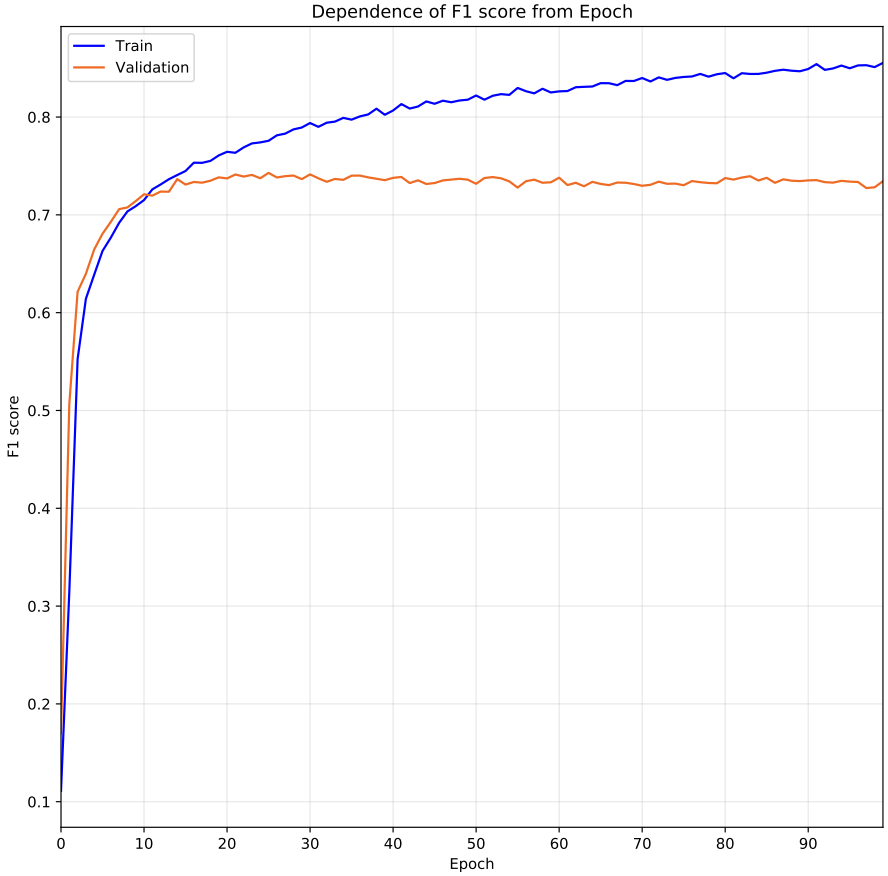

Рис. 6: Зависимость f1 меры на обучающей и валидационной выборках от эпохи (для классификации аспектных категорий на уровне предложений).

В таблице 4 представлены значения метрик качества для построенного классификатора и контрольного алгоритма на тестовой выборке.

Таблица 4: Метрики качества классификации на тестовой выборке.

Классификатор	Precision	Recall	F1
Baseline	-	-	0.55882
Bi LSTM	0.73703	0.76438	0.75046

4.2 Извлечение и классификация аспектных терминов

Так как на обучающей выборке аспектные термины в основном принадлежат только одной аспектной категории одновременно[13], то сделано допущение, что аспектный термин может иметь только одну категорию. Таким образом, вместо того, чтобы рассматривать задачу как задачу классификации на пересекающиеся классы, она рассматривалась как задача многоклассовой классификации.

Задачи извлечения аспектных терминов и определения их категории были объединены в одну, путем добавления к множеству аспектных категорий метки, обозначающей, что слово не является частью аспектного термина.

Для классификации так же использовалась Bi LSTM, но в качестве ответа выбирался один класс с наибольшим значением. В качестве функции потерь использовалась перекрестная энтропия (32), где $y^* \in \{0, 1, \ldots, n\}$[14].

$$loss(y, y^*) = -log\frac{e^{y_{y*}}}{\sum_{i=0}^{n} e^{y_i}} = -y_{y^*} + log\sum_{i=0}^{n} e^{y_i} \tag{32}$$

На рис. 7 изображена зависимость f1-метрики, вычисляющейся для отдельных слов. В качестве оптимального значения выбрано 50 эпох.

Затем последовательно идущие слова, для которых предсказана одна и та же аспектная категория объединяются в один аспектный термин. Когда явные аспектные термины для предложения определены, они сравниваются с предсказанными аспектными категориями предожения, определенными предыдущим классификатором. Если для предложения заявлена аспектная категория, но среди явных аспектных терминов ее нет, то считается, что она неявная и в список *мнений* предложения добавляется новое мнение, в котором на месте аспектного термина указано *NULL*.

В таблице 5 представлены значения метрик качества для построен-

[13]В обучающей выборке имеют только одну категорию 2988 аспектных терминов (исключая неявные), 2 - 79, 3 - 2.

[14]Здесь неаспектный термин имеет порядковый номер 0.

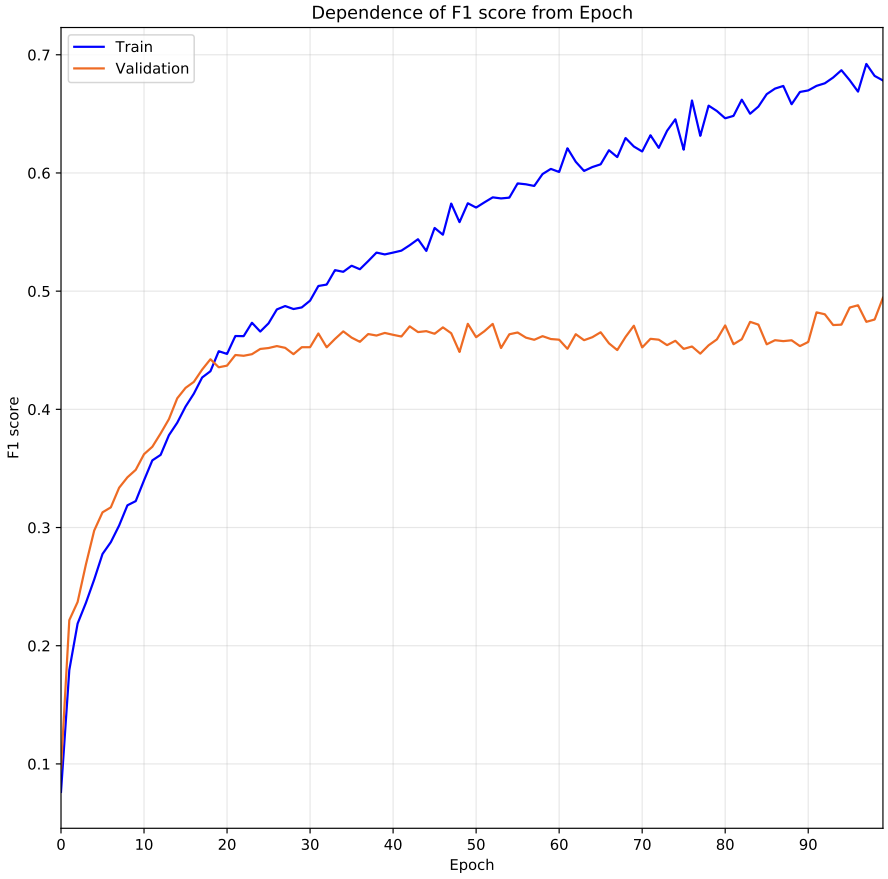

Рис. 7: Зависимость f1 меры на обучающей и валидационной выборках от эпохи (для классификации аспектных категорий на уровне отдельных слов).

ного классификатора и контрольного алгоритма на тестовой выборке.

Таблица 5: Метрики качества классификации на тестовой выборке.

Классификатор	Precision	Recall	F1
Baseline	-	-	0.39441
Bi LSTM	0.60308	0.36825	0.45728

Глава 5. Определение тональности

5.1 Архитектура нейронной сети

Для определения тональности аспектных терминов так же как и для их извлечения использовалась Bi LSTM. Полученные на выходе скрытые состояние передавались в Tree LSTM, реализованную с помощью библиотеки DGL.

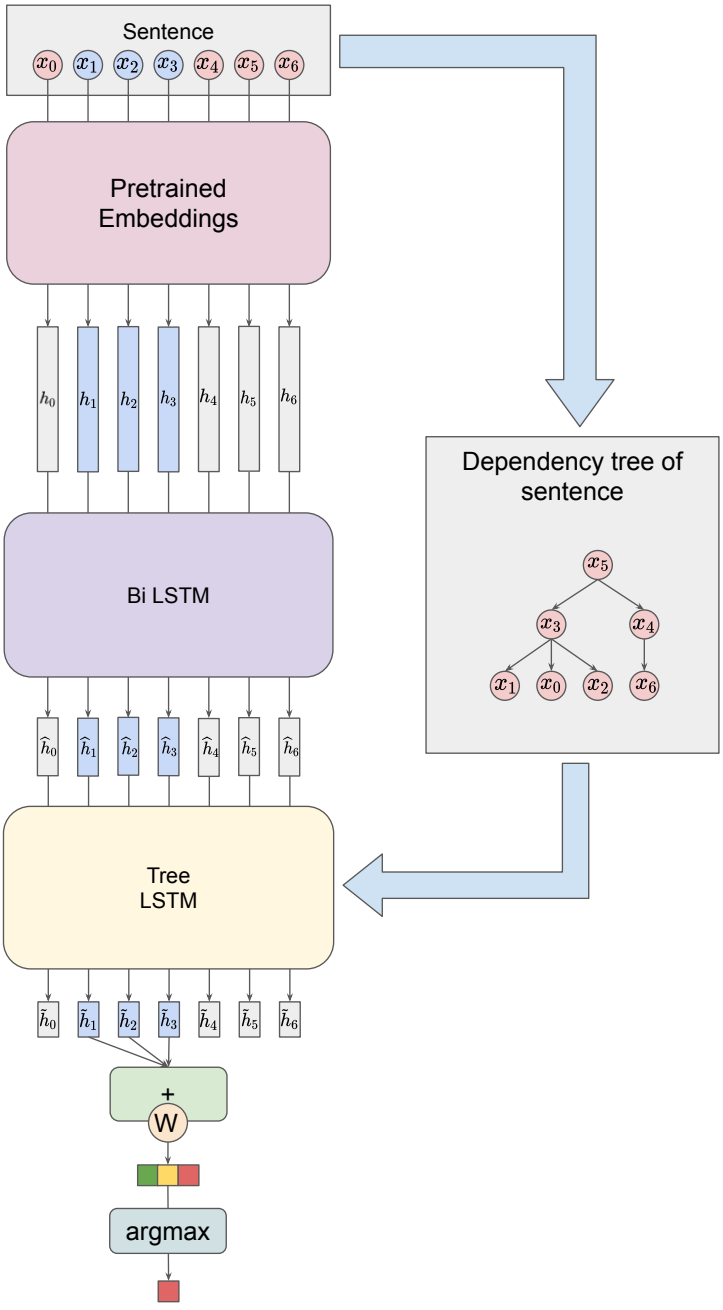

Рис. 8: Схема определения тональности аспектного термина.

В отличие от предыдущих классификаторов данный формирует отдельные батчи для каждого аспектного термина. Элемент батча содержит индексы векторных представлений слов предложения, маску аспектного термина, граф - дерево зависимостей предложения.

По полученному батчу нейросеть формирует тензор из векторных представлений слов. Они проходят через слой дропаута и подаются на вход Bi LSTM слою. Полученные на выходе скрытые состояния, соответствующие элементам последовательности, проходят через очередной слой дропаута и подаются в Tree LSTM слой.

Деревья зависимостей предложений имеют различную форму, из-за чего параллелизация вычислений Tree LSTM слоя нетривиальная задача. Однако DGL предоставляет API для обмена сообщениями между смежными вершинами графа. В качестве сообщений выступают аттрибуты вершин и ребер графа.

Деревья зависимостей объединяются в один граф, изменения индексов вершин и ребер происходит в соотвествии с таблицей 6. Векторные представления слов, полученные из Bi LSTM, представляют собой тензор размерности $[B \times T \times E]$, где B - количество предложений в батче, T - длина самого длинного предложения, E - размерность векторного пространства слов. Этот тензор преобразуется в тензор размерности $[\sum_{i=0}^{n} T_i \times E]$. Индексы векторов слов в тензоре соотвествуют вершинам графа.

Таблица 6: Правило изменения индексов вершин и ребер графов.

	Граф G_0	Граф G_1	...	Граф G_n
Изначальный индекс	$0,$ $1,$ $\dots,$ T_0	$0,$ $1,$ $\dots,$ T_1	...	$0,$ $1,$ $\dots,$ T_n
Новый индекс	$0,$ $1,$ $\dots,$ T_0	$T_0,$ $T_0 + 1,$ $\dots,$ $T_0 + T_1$	...	$T_0 + T_1 + \dots + T_{n-1},$ $T_0 + T_1 + \dots + T_{n-1} + 1,$ $\dots,$ $T_0 + T_1 + \dots + T_{n-1} + T_n$

Производится топологическая сортировка вершин графа, множество вершин графа разбивается на непересекающиеся подмножества вершин, не связанных отношением частичного порядка. Последовательно для каждого множества вершин параллельно выполняются функции *message*, *reduce*, *apply*.

- В функции *message* от начальной к конечным вершинам передаются скрытое состояние h и состояние ячейки памяти c.
- В функции *reduce* вычисляется среднее полученных скрытых состояний (22), значение фильтра забывания (26) и второе слагаемое (27).
- В функции *apply* вычисляется вектор новых значений ячейки памяти (24) и фильтров входа (23), выхода (25). Затем обновляется состояние ячейки памяти (27) - произведение Адамара вектора новых значений ячейки памяти и входного фильтра складывается с полученным в *reduce* вторым слагаемым. Затем вычисляется скрытое состояние узла (28).

После прохождения по всем множествам вершин получаем скрытые состояния для каждого слова. С помощью масок аспектных терминов из скрытых состояний выбираются векторы аспектных терминов. Скрытые состояния слов одного аспектного термина усредняются и подаются в полносвязный слой, осуществляющий линейное преобразование (29), где n - количество тональностей, т.е. равен 3. В качестве ответа классификатора выбирается класс с максимальным значением.

5.2 Обучение сети

В качестве функции потерь в процессе обучения используется перекрестная энтропия (32).

На рис. 9 отображены зависимость точности на обучающей и валидационной выборках от эпохи обучения. Как видно на графике, максимальная точность на валидационной выборке достигается в районе 30 эпох, а затем падает, т.е. начинает переобучаться.

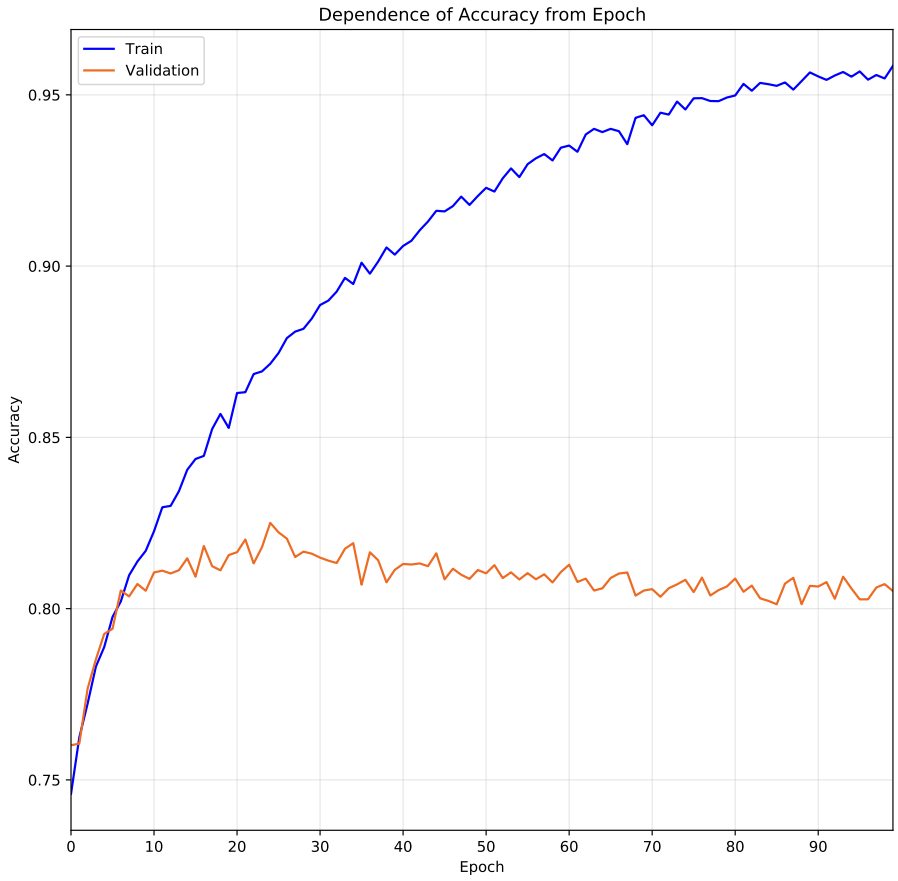

Рис. 9: Зависимость точности на обучающей и валидационной выборках от эпохи.

Размерность скрытого состояния в BiLSTM равна 2*50, в Tree LSTM 30. Первый слой дропаут (перед BiLSTM) имеет вероятность 0.7, второй (перед Tree LSTM) - 0.1.

Нейронная сеть с предложенной архитектурой превосходит результаты контрольньного алгоритма и показывает на тестовой выборке следующий результат (таблица 7).

Таблица 7: Метрики качества классификации на тестовой выборке.

Решение	Accuracy
baseline	0.71
Bi LSTM + Tree LSTM	0.80769

Заключение

В данной работе было проведено исследование методов машинного обучения для задачи аспектно-ориентированного анализа отзывов на русском языке. Рассмотрены существующие методы извлечения аспектных терминов и определения тональности.

Построен конвейер для решения задачи аспектно-ориентированного анализа, который состоит из следующих этапов[15]:

- предобработка текстов
  - разбиение на токены
  - проверка орфографии
  - построение деревьев зависимости предложений
- извлечение и классификация аспектных терминов
  - определение аспектных категорий предложения
  - определение аспектных категорий отдельных слов
  - обединение результатов классификаторов для определения явных и неявных *мнений*
- определение тональности аспектного термина

Описана работа рекуррентных нейронных сетей. Предложена и реализована архитектура нейронной сети, состоящей из Bi LSTM и Tree LSTM слоев, для определения тональности аспектного термина. Построенные классификаторы превосходят метрики качества контрольных алгоритмов.

[15]Исходный код проекта доступен в репозитории: https://gitlab.com/davydovdmitry/absa

Список литературы

[1] «88% Of Consumers Trust Online Reviews As Much As Personal Recommendations». https://searchengineland.com/88-consumers-trust-online-reviews-much-personal-recommendations-195803

[2] «Исследование: влияние отзывов на мнение потребителя». https://vc.ru/marketing/91417-issledovanie-vliyanie-otzyvov-na-mnenie-potrebitelya

[3] Рой Д.А., Ефремова Н.Э. «Методы извлечения аспектных терминов из мнений». Новые информационные технологии в автоматизированных системах. 2018.

[4] Turney P. «Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classifi cation of Reviews». 2002. pp. 417-424.

[5] Блинов П., Котельников Е. «Семантическое сходство в задаче аспектно-эмоционального анализа».

[6] Ramshaw L., Marcus M. «Text chunking using transformation-based learning». Natural language processing using very large corpora, Springer Netherlands, 1999, pp. 157-176.

[7] Chernyshevich M. «IHS R&D Belarus: Cross-domain Extraction of Product Features using Conditional Random Fields». Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp.309–313.

[8] Jakob N., Gurevych I. «Extracting opinion targets in a single- and cross-domain setting with conditional random fields». In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. 2010. pp. 1035-1045.

[9] Lafferty J., McCallum A., Pereira F. «Conditional random fields: probabilistic models for segmenting and labeling sequence data».

Proceedings of the Eighteenth International Conference on Machine Learning. 2001. pp. 282–289.

[10] Четвёркин И., Лукашевич Н. «Автоматическое извлечение оценочных слов для конкретной предметной области». 2010.

[11] Sowjanya M., Srividya K. «Aspect Based Sentiment Anaysis using POS Tagging and TFIDF».

[12] Popescu A., Etzioni O. «Extracting Product Features and Opinions from Reviews»

[13] Rana T., Cheah Y. «Improving Aspect Extraction Using Aspect Frequency and Semantic Similarity-Based Approach for Aspect-Based Sentiment Analysis». Recent Advances in Information and Communication Technology 2017, Advances in Intelligent Systems and Computing 566.

[14] (Rubenstein H., Goodenough J. «Contextual correlates of synonymy». Communications of the ACM 8(10). 1965. pp. 627-633.

[15] Pantel P. «Inducing ontological cooccurrence vectors». Proceedings of the 43rd Conference of the Association for Computational Linguistics. Association for Computational Linguistics. 2005. pp. 125-132.

[16] Sahlgren M. «The Distributional Hypothesis. From context to meaning». Distributional models of the lexicon in linguistics and cognitive science (Special issue of the Italian Journal of Linguistics), Rivista di Linguistica. 2008. pp. 33-53.

[17] Mikolov T., Chen C., Dean J., Corrado G., «Efficient Estimation of Word Representations in Vector Space». 2013.

[18] Теслец Я. «Введение в общий синтаксис». 2001.

[19] Шаров С. «Средства компьютерного представления лингвистической информации». 1996.

[20] Апресян Ю. «Лингвистический энциклопедический словарь». / Главный редактор В. Н. Ярцева. — М.: Советская энциклопедия, 1990. pp. 685.

[21] Алпатов В. «О грамматике состовляющих и грамматике зависимостей».

[22] Strazny P. «Encyclopedia of Linguistics». 2005. pp. 397—401.

[23] Robinson J. «Dependency structures and transformation rules». Language 46. 1970. pp. 259–285.

[24] Hays D. «Dependency theory: A formalism and some observations». 1964. pp. 511–525.

[25] Gaifman H. «Dependency systems and phrase-structure systems». Information and Control 8(3). 1965. pp. 304–337.

[26] Debusmann R. «An Introduction to Dependency Grammar». 2000.

[27] Nivre J., de Marneffe M., Ginter F., Goldberg Y., Hajič J., Manning C., McDonald R., Petrov S., Pyysalo S., Silveira N., Tsarfaty R., Zeman D. «Universal Dependencies v1: A Multilingual Treebank Collection». Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). 2016. pp. 1659-1666.

[28] Peng Qi, Timothy Dozat, Yuhao Zhang and Christopher D. Manning. «Universal Dependency Parsing from Scratch». In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 2018. pp. 160-170.

[29] Nivre J. «Towards a Universal Grammar for Natural Language Processing». 2015.

[30] Hochreiter S., Schmidhuber J. «Long Short-Term Memory». Neural Computation 9(8). 1997. pp. 1735–1780.

[31] Sun K., Zhang R., Mensah S., Mao Y., Liu. «Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree». Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019. pp. 5679–5688.